

Blogg

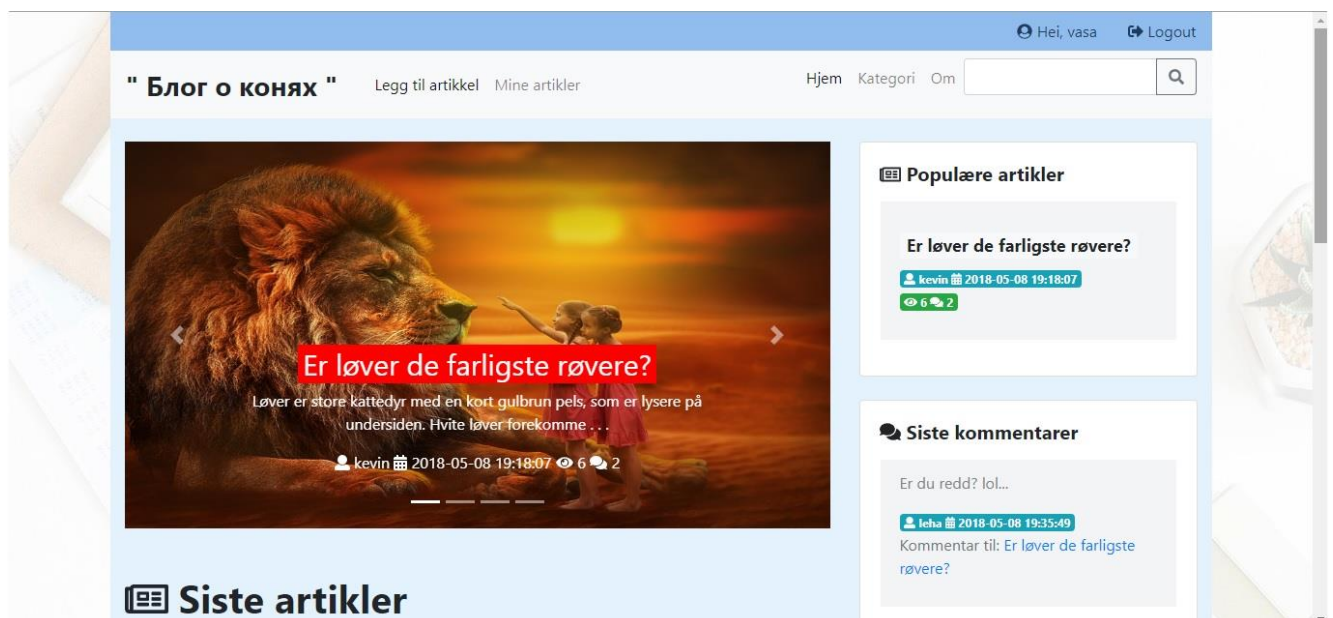
Avdeling for teknologi

ITE1805-1 18V Databaser og webapplikasjoner 1

Gruppe 7

Aleksei Poroshin

Georgii Bogdanov



Narvik

Innhold

Forord.....	3
Sammendrag	4
1. Prosjektoppgave	5
1.1 Nødvendig informasjon.....	7
2. Utseende og beskrivelse av bloggen	7
3. Database.....	10
4. Beskrivelse av valgt løsning	10
5. Hva har vi lært	12
Literaturliste	13

Vedlegg

Forord

Denne rapporten er en del av ITE1805-1 18V Databaser og webapplikasjoner 1. Oppgaven var går ut på å utvikle en blogg. Løsningen skal baseres på bruk av HTML 5, CSS, PHP og evt. klient JavaScript. Bloggen skal benytte en database for lagring av informasjon.

Vår Gruppe №7 består av to studenter: Aleksei Poroshin og Georgii Bogdanov. Løsningen som ble brukt i vårt prosjekt ble valgt i samsvar med kravene i oppgaven, materialet som ble studert i dette faglige semesteret.

Fagbetegnelse: ITE1805-1 18V	Gruppenavn: «Gruppe 7»	Innleveringsdato: 08.05.2018
	Tittel: «Kryptovaluta i et Samfunnsperspektiv (ved eksemplet på Bitcoin).	
Semester: Vår 2018	Veileder: Knut Collin	Antall sider:
Arbeidet avsluttet: 08.05.2018		
Kandidater:		
1. Aleksei Poroshin 2. Georgii Bogdanov		

Sammendrag

Prosjektoppgave – Blogg går ut på å utvikle en blogg. Løsningen skal baseres på bruk av HTML 5, CSS, PHP og evt. klient JavaScript. Bloggen skal benytte en database for lagring av informasjon.

1. Prosjektoppgave

Oppgaven går ut på å utvikle en blogg. Løsningen skal baseres på bruk av HTML 5, CSS, PHP og evt. klient JavaScript. Bloggen skal benytte en database for lagring av informasjon. MySQL (MariaDb) på kark.hin.no benyttes til dette. Det er ønskelig at bloggen har følgende funksjonalitet:

Krav til bloggfunksjonalitet:

1. Løsningen skal ha støtte for mange blogger. En registrert bruker skal kunne opprette sin egen blogg og bli eier av denne.
2. Hovedsiden for en blog bør vise alle, evt. et utvalg av alle blogginnlegg sortert på dato med antall kommentarer.
3. Eieren av bloggen skal måtte logge seg på for å legge inn nye innlegg.
4. Anonyme brukere skal kunne lese innlegg.
5. Brukere må kunne registrere seg med nødvendig informasjon. Passord for brukere bør lagres med bruk av hash i databasen. (Benytte php metodene `password_hash()` og `password_verify()`). Du bestemmer selv annen relevant informasjon som bør lagres for en bruker.
6. Ved registrering bør det sendes epost til brukeren for bekreftelse av epostadressen, bekreftelse er nødvendig før konto tas i bruk. Notatet [Sending av epost for prosjektoppgaven](#) inneholder beskrivelse av hvordan sende epost.
7. Det bør være mulig å kommentere innlegg og kommentarer. Kun registrerte bruker skal kunne gjøre dette.
8. Du bestemmer selv formatet for et blogg innlegg, hvilke elementer som skal være med. Naturlige elementer vil være: Tittel, tekst for innlegget, dato, forfatter, stikkord.
9. Et blogginnlegg bør kunne ha vedlegg av begrenset størrelse. Dette kan for eksempel være bilder. Vedlegg lagres i databasen og vises ved lesing av innlegget. Typen vedlegg og størrelsen på disse bør begrenses.
10. Et blogginnlegg bør kunne ha en eller flere tags (stikkord). Det bør være mulig å hente ut alle innlegg basert på tag, nye tags bør kunne opprettes.
11. Eldre innlegg bør kunne aksesseres på månedsbasis, slik at det er lett å få opp innlegg for en aktuell måned, oversikten bør inneholde antall innlegg denne måneden.
12. Det kan være treftellere på de ulike innleggene som inkrementeres ved aksess og vises på web.

13. Det bør være mulig å søke på innleggenes emne og innhold for å finne det en er på jakt etter.
14. Eieren av bloggen bør kunne drive vedlikehold av bloggen og foreta operasjoner som sletting av upassende kommentarer.
15. Sletting av kommentarer skal logges i egen loggtabell som viser den slettede kommentaren samt tidspunkt (dato og klokkeslett) for slettingen, bruk med fordel en trigger her.

Krav til database.

Databasen blir fundamentet i bloggen og må være normalisert opp til 3NF. Det anbefales å bruke WorkBench til modellering og konstruksjon av databasen. Databasen må modelleres med relasjoner, primærnøkler og fremmednøkler etter prinsipper for ER-modellering. Databasen må ha støtte for all funksjonalitet nevnt under "**Krav til bloggfunksjonalitet**". Det vil si at datamodellen kan benyttes for å lage en komplett løsning som spesifisert, som så via PHP-kode benytter seg av den forrigling og funksjonalitet databasen gjennom dette tilbyr. Datamodellen er en del av vurderingsgrunnlaget.

Generelle krav til løsningen.

- Løsningen bør fungere i de mest vanlige nettleserne på ulike plattformer.
- Alle sider bør benytte samme stilark slik at stil/utseende blir ensartet. JavaScript kan benyttes ved behov
- Løsningen må ha beskyttelse mot SQL injection og HTML/JavaScript i innlegg og kommentarer.
- HTML og CSS koden skal være i henhold til W3C's anbefalinger.
- Benytt objektorienterte prinsipper ved PHP kodingen.
- Sesjonsdata bør beskyttes.
- I MySQL skal tabelltypen være INNODB med støtte for fremmednøkler, referanseintegritet skal ivaretas.
- HTML og PHP koden bør adskilles ved bruk av et template system ala Twig.

NB ! All kode som benyttes skal være egenprodusert. I de tilfellene der det benyttes kode skrevet av andre skal dette tydelig opplyses med referanse. Kildehenvisninger er meget viktig. Hver student må fylle ut egenærklæring angående egen kode og bruk av kilder. Egenerklæringen er obligatorisk og ligger her: [Egenerklæring prosjektoppgaven](#)

GIT

All kode skal under hele prosjektperioden ligge på GITLab løsningen hos UIT, adresse <https://source.uit.no/> (Lenker til en ekstern side.)Lenker til en ekstern side.. GIT skal benyttes aktivt av alle prosjektmedlemmer i utviklingen slik at historikken vil vise hvem som

har gjort hva og når til en hver tid. Faglærere må gis tilgang til repository fra starten av prosjektet, ast175@uit.no og kco006@uit.no må gis Reporter access til prosjektet.

1.1 Nødvendig informasjon

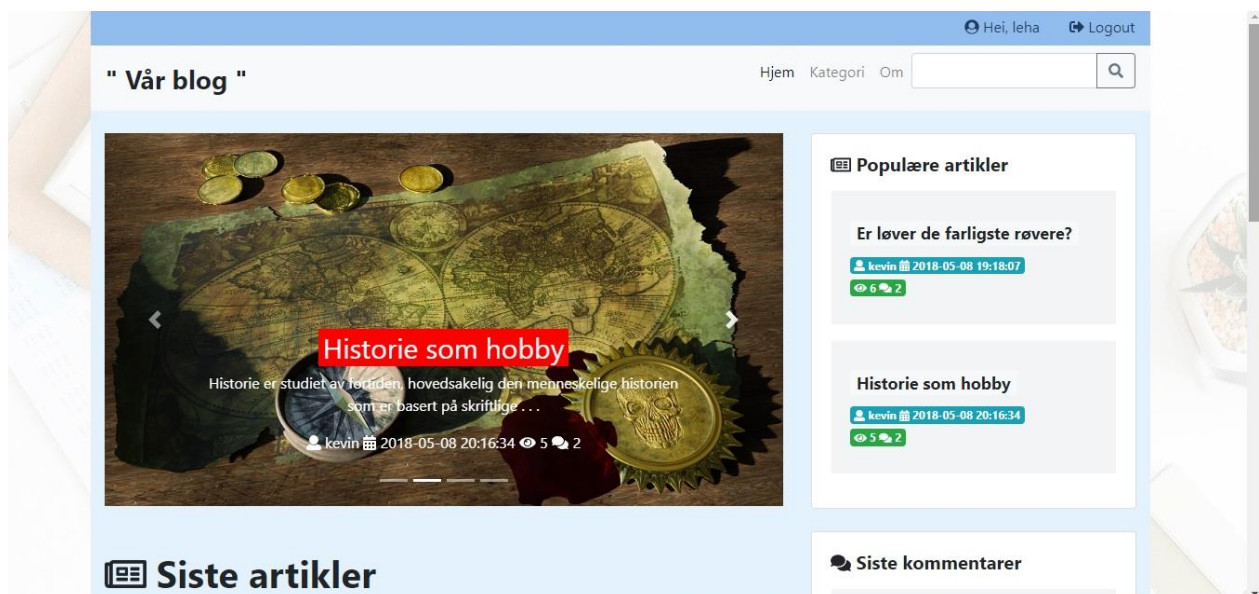
Link til Bloggen: <https://kark.uit.no/~apo033/ProsjektoppgaveGruppe7/>

Autorisasjonsdata for testbrukere er i Tabell 1.

Login	Passord	Beskrivelse
kevin	12345	har egen blogg
vasa	kotik	har egen blogg
john	qwerty	bruker uten blog
leha	111111	bruker uten blog

Tabell 1 autorisasjonsdata for testbrukere

2. Utseende og beskrivelse av bloggen



Figur 2 Hjemmeside av bloggen

Funksjonen til bloggen er utformet helt i samsvar med oppgaven. Løsningen støtter mange blogger. Bloggen støtter tre typer av brukere: en uautorisert bruker, en autorisert bruker uten blogg og en autorisert bruker som laget sin blogg.

Uautorisert bruker: Denne brukeren er tilgjengelig all informasjon (om bloggere, alle deres skriftlige artikler, kommentarer) i lesemodus.

Autorisert bruker uten blogg: Denne brukeren er tilgjengelig all informasjon (om bloggere, alle skriftlige artikler, kommentarer). Han hadde også mulighet til å skrive kommentarer og svare på kommentarer, lage sin egen blogg, slette sin konto.

Autorisert bruker som laget sin blogg: Denne brukeren er tilgjengelig all informasjon (om bloggere, alle skriftlige artikler, kommentarer). Han har mulighet til å skrive kommentarer og svare på kommentarer i alle blogger, slette sine kommentarer og kommentarer fra andre brukere på bloggen hans, slett sin artikkel, slett alle artiklene hans, slett bloggen sin, slett sin konto.

Hovedsiden for en blogg viser:

Autorisasjonsblokk

- Avhengig av autorisasjonsstatusen den inneholder de nødvendige knappene for: Login (Kaller det modale vinduet for autorisasjon), Signup (Kaller det modale brukerregistreringsvinduet), Knapp med brukernavn (gå til brukerens profilside eller blogginnehaverens side), logout.

Navigasjonsblokk:

- Inneholder navnet på bloggen, navigasjonsknappene på nettstedet, nettstedssøk. Brukeren som opprettet sin blogg vil motta flere knapper: legg til artikkel, Mine artikler.

Innholdsblokk:

- **(Hovelside)** Inneholder en karusell med fem populære innlegg, siste artikler, Informasjon om hovedblogg, Liste over bloggere, liste stikkord.

(Dette er hovedvinduet for visning av innhold, det endres avhengig av om brukeren er valgt.)

Sidebar:

- Inneholder populære artikler, siste fem kommentarer, artikler arkiv (innlegg kun aksesseres på månedsbasis).

Footer:

- Hardkodet informasjon om prosjektoppgaven.

Andre sider

Kategori:

- Inneholder en liste over kategorier av artikler, når du velger en kategori, går brukeren til siden med alle artiklene i denne kategorien.

Om:

- Avhengig av autorisasjonstilstanden den inneholder: Informasjon om Vår blog (Hovedblogg), konto kontrollpanel.

Legg til artikkel:

- Denne siden er bare tilgjengelig for brukere som har en blogg. Inneholder et skjema for å lage en artikkel.

Mine artikler:

- Denne siden er bare tilgjengelig for brukere som har en blogg. Viser alle artikler av eieren av bloggen, med muligheten til å slette alle artikler med en knapp.

Artikkel:

- Viser artikkelen (Tittel, forfatter, dato, antall visninger, tekst, bilde), kommentarer, et skjema for å legge til kommentarer (avhengig av autorisasjonstilstanden), en knapp for å svare på kommentarer (avhengig av autorisasjonstilstanden), en artikkel slettingsknapp (for blogginnehaveren).

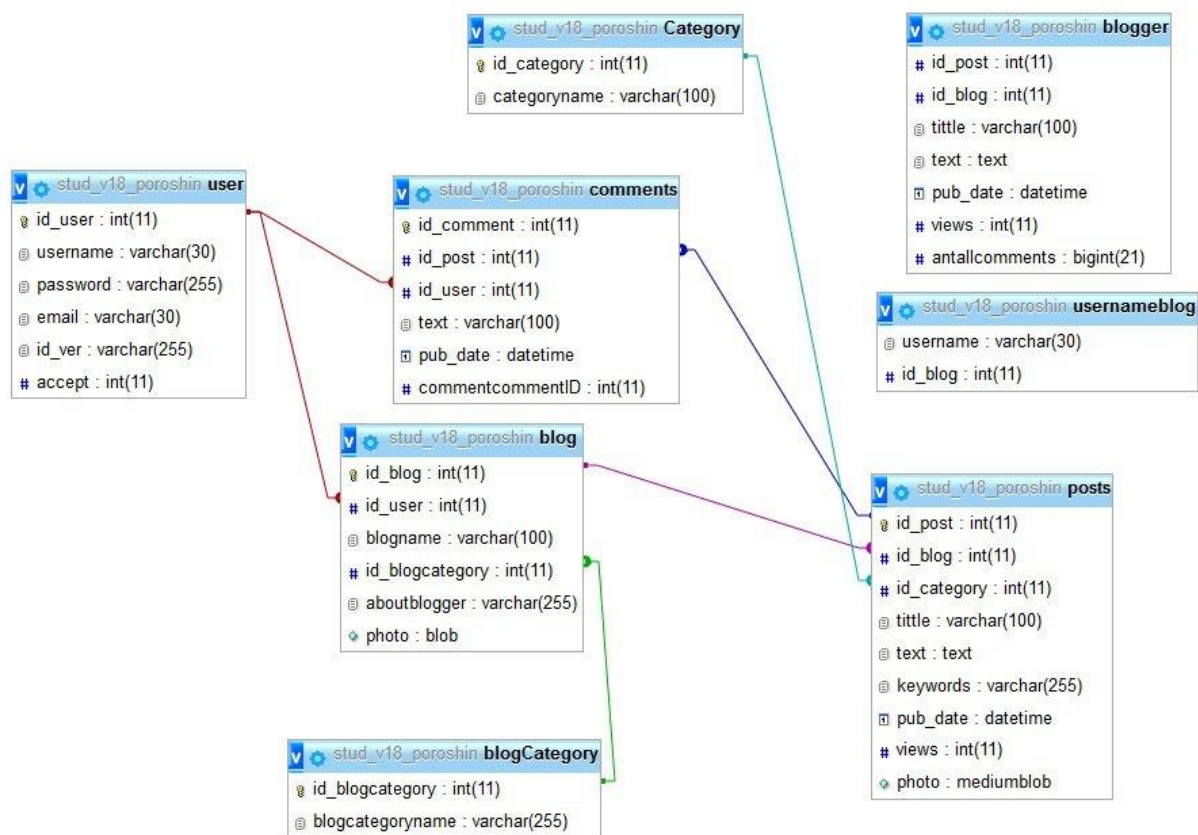
Profilside:

- Denne siden er bare tilgjengelig for autoriserte brukere. Brukerens profilside eller blogginnehaverens side. Administrer bloggen og konto.

Fra hjemmesiden for alle typer brukere, kan du få følgende informasjon: Informasjon om bloggere, antall artikler som er sett, publiseringsdato og mer.

database

3. Database



Figur 4 Datamodellen

4. Beskrivelse av valgt løsning

Løsningen består av hovedfilen index.php som filer med tilleggsfunksjoner er koblet til (lasting av bilder fra databasen, handlinger når du trykker på autorisasjonsknappene, registrering, en konfigurasjonsfil for tilkobling til databasen, etc.). Også objektorienterte prinsipper ble brukt, det er klasser for objekter - en blogg, kommentarer, artikler, blogg. En klasse med grunnleggende funksjoner for tilgang til databasen Blog.class.php og «interface» klass til det er opprettet. Twig «templates».

4.1 Arbeider med databasen

Alle forespørsler til databasen er skrevet inn Blog.class.php ved hjelp av PDO på samme måte som vi studerte.

Der blir alle de nødvendige forespørsler gjort, for eksempel: valg fra databasen, oppdatering, tillegg, sletting. Mest funksjonell av bloggen er gjort ved hjelp av spørringer i databasen, som det var praktisk.

```
public function lastFiveComments(): array
{
    $lastfivecomments = array();

    try {
        $stmt = $this->db->prepare( statement: "SELECT c., u.username, p.tittle as posttittle FROM `comments` as c,
        user as u inner join posts as p WHERE u.id_user = c.id_user and c.id_post=p.id_post GROUP by c.pub_date desc limit 5 ");
        $stmt->execute();

        while ($comments = $stmt->fetchObject( class_name: "Comments")) {
            $lastfivecomments[] = $comments;
        }
    } catch (InvalidArgumentException $e) {
        print $e->getMessage() . PHP_EOL;
    }
    return $lastfivecomments;
}
```

Figur 4 Eksempelforespørsel i databasen

4.2 Objektorienterte prinsipper

Vi brukte objekter til å utveksle data mellom hovedprogrammet og databasen. Klasser for objekter - en blogg, kommentarer, artikler, blogg.

4.3 Kodeseparasjon

For separasjon av PHP og HTML-kode Twig ble brukt. I henhold til dokumentasjonen ble Twig-templates opprettet, i hvilke data ble sendt i form av arrayer og tilleggsvariabler for visning i nettleseren. Twig-templates bruker standardkoden HTML, betingede setninger, looper, etc. fra TWIG. For å designe av sidene ble det brukt en «Bootstrap» og «Awesome» fonter.

```
//Eldre innlegg aksesseres på månedsbasis
if (!empty($_GET['month'])) {
    $arkhivvalg = $blogdb->arkhivValg($_GET['month']);

    echo $twig->render( name: 'allusersposts.twig', array('blogname' => $_SESSION['blogname'], 'period' => $_GET['month'],
    'arkhivvalg' => $arkhivvalg, 'arkhiv' => $arkhiv, 'lastfivecomments' => $lastfivecomments, 'topfiveposts' => $topfiveposts,
    'innlogget' => $_SESSION['innlogget'], 'bloggernavn' => $_SESSION['name'], 'blogowner' => $_SESSION['blogowner']));
}
```

Figur 5 Eksempel Utdata Template Twig

Literaturliste

- 1) Databasesysteme, Bjørn Kristoffersen
- 2) Wikipedia (2018). <https://www.no.wikipedia.org/>
- 3) <https://stackoverflow.com/>
- 4) <http://getbootstrap.com>
- 5) <https://www.w3schools.com>
- 6) <https://twig.symfony.com/>
- 7) <https://pixabay.com>

Vedlegg

dagbokAleksey.pdf

dagbokGeorgii.pdf